DESIGN AND DEVELOPMENT OF A CONTOUR PLOTTING PACKAGE

by

SHAKTI KAPOOR

DEPARTMENT OF MECHANICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

JUNE, 1982

DESIGN AND DEVELOPMENT OF A CONTOUR PLOTTING PACKAGE

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of
MASTER OF TECHNOLOGY

by

SHAKTI KAPOOR

to the

DEPARTMENT OF MECHANICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KANPUR

JUNE, 1982

:Y
Acc. No. A 82649

POST GRADUATE OFFICE
Submitted on 28.6.82

CERTIFICATE

CERTIFIED that the thesis entitled 'Design and Development of a Contour Plotting Package' has been carried out under my supervision and it has not been submitted elsewhere for a degree.

Kanpur

23 June 1982

Dhande
Dr. S.G. Dhande
Assistant Professor
Mechanical Engineering Department
INDIAN INSTITUTE OF TECHNOLOGY,
KANPUR-208016, U.P.

ACKNOWLEDGEMENTS

I would like to thank Dr. S.G. Dhande for his guidance and constant encouragement. It was he who suggested this problem to me. But for his invaluable suggestions at various stages of the project I would not have finished my work in time.

I also thank Mr. Abhijit Bhattacharyya (B.Tech.) for his useful comments in the earlier stages and then critically reviewing the programme in order to decrease redundancies.

My thanks are due to Mr. D.P. Saini for excellent typing. I must not forget to express my thanks to Mr. S.S. Kushwaha for making the excellent diagrams.

SHAKTI KAPOOR

CONTENTS

LIST OF FIGURES

ABSTRACT

In the present work, a technique for plotting contours of constant function values over a grid of rectangular cells has been proposed and developed in the form of a computer program. Input data needed to draw the contours consists in the form of function values at all nodes of the grid. The grid is considered to be enclosed by a rectangular boundary. The procedure consists of locating appropriate contour points on the grid for a particular function value and then generating a smooth curve passing through the contour points. The method developed can take into account not only closed and open contours but can also draw multi-branched contours. Further, some typical cases of degeneracies can be dealt with using the proposed approach. Two illustrative cases have been tested and the contour plots have been presented.

However, the present work does not resolve successfully all the possible cases of degeneracies.

Chapter 1

INTRODUCTION

1.1 An Overview of Contour Plotting Applications

A common problem of graphics encountered by designers and scientists is that of representing data in the form of a set of functional values of a parameter which, in turn is a function of two variables. The usual methods of representing a function of two variables are contour plots and planar projections.

While planar projections [1] depict the function in such a manner that its properties are most easily visualised, contour plots on the other hand emphasize quantitative aspects also : even numerical information can be extracted from the plots. So planar projections are used more for qualitative visualisation and contour maps are for quantitative representation of functions of two variables of the type $Z = Z\ (x,y)$.

Contour maps most commonly seen are representation of various quantities like elevation, temperature, pressure as a function of geographic position.

Other examples where contour maps find applications are : Streamlines or constant potential lines or constant vorticity lines as seen in any fluid flow situation ; constant temperature lines as seen in any heat transfer problem. A more general example is that of contour maps used for plotting of equal loudness curves drawn as a function of the intensity and frequency of an audible tone [2].

In most applications of contour maps the relationship between the dependent variable and the independent variables cannot be conveniently expressed by a closed-form equation. However there are some applications in which contour maps are used even though an equation is readily at hand. This is because the contour maps facilitate visualisation of data. An example of the latter is a plot of the equipotential lines around an electric dipole.

This apart, many problems can be solved using the techniques for processing contour maps [4].

1.2 Methodologies of Drawing Contours

The problem of plotting contours [3] can be stated as follows : Let $w = w(x,y)$ be a two variable function defined in a rectangular domain R ($a \leq x \leq b$, $c \leq y \leq d$). Now let us suppose that $w = w(x,y)$ is

known at the nodes of a regular grid traced on R, and that it is required to plot n contour lines : w (x,y) = C_k, k = 1,.....,n. Denoting each node by (i,j), a matrix w (i,j) can be defined. The cell of the grid is the rectangle with four adjacent nodes of the grid as vertices. The edge of the cell which is intersected by the particular contour line is said to be the potential element (edge) for that particular contour value.

Two classes of methods are available for finding contour points lying on the grid and then joining them. In the first class, all the intersection points are found first and then with some criterion these points are joined. In the methods belonging to the second class one point of the contour is searched, and the contour is followed until it stops.

Most of the workers in this area have found sequential cell search method better than the other method. This is because the ordering process in the first method requires very large storage capacity and is very slow. The methods belonging to the second class differ when it comes to resolving degenerate cases (refer Chapter 3). Sometimes triangular grids are formed in order to obviate the problem of degeneracy.

The present work has tried to deal with some typical cases of degeneracy.

1.3 Scope of the Present Work

The present work has been designed and developed to draw contours for those cases where data is available at a set of grid nodes formed by regular spacing of grid lines. In other words, cells of the mesh formed by grid lines are squares. A separate subroutine is also available for those cases where spacing between the grid lines is irregular.

The package provides two options to the user for locating contour points lying on the grid lines. These points are found by either using linear or cubic interpolation schemes on the potential edges of the cell.

In chapter 2 a description of the general methodology of contour plotting has been given. Chapter 3 describes how degenerate situations are dealt with and thus how multi-branch contours can be drawn without re-iterating a contour or getting into an infinite loop. Illustrative examples have also been included. Two sets of data have been used to test the programme. Discussion on the case studies has been included in Chapter 4. The results of the case studies and guidelines for further work have been summarized in Chapter 5.

Chapter 2

PROPOSED METHODOLOGY OF CONTOUR PLOTTING

2.1 Summary of the Procedure

Given function values at grid nodes of rectangular grid the method of plotting of contours involves the following basic steps:

Step 1: Given a contour value the first step is to find any point on a grid line which has this contour value.

Step 2: Having found any one point which lies on the contour, the next step is to search the other edges of the cell surrounding the edge on which above-found point lies, for the possibility of other contour points. This transpires from the simple fact that once the contour enters the cell from any one of the four edges of the cell, it should then come out intersecting any one of the three remaining edges of the cell. Unless, of course, the contour reaches the boundaries of the grid or else it ends abruptly. The latter being very

unlikely. Only maximum and minimum contours can end abruptly [4]. It has been tacitly assumed that the function increases or decreases monotonically within a cell.

Step 3 : For finding the exact location of the contour on the edge of the cell, various interpolation methods can be used.

Step 4 : Having found all the points at which the contour intersects the grid, a smooth curve is fitted through these points.

Step 5 : The next step is to display these contours on a graphics terminal.

In the next section each of these steps have been taken up separately so as to point out how they can be implemented.

2.2 Detailed Description of the Procedure

Step 1: Whatever be the contour, it will intersect horizontal grid lines or at the most lie on it. So it becomes sufficient if only the horizontal grid lines are scanned for potential elements, so as to create all possible loops of the considered contour. In fact one goes through all horizontal grid lines and stores all potential elements on them with a label indicating horizontal

grid line index. The left and right vertical grid boundaries potential elements are also stored. The potential element is identified by its lower co-ordinate value.

Once potential elements have been stored it will be advantageous if a set of rules is followed in choosing the first potential element from the store. Full implications of these rules will be brought out at the end of the next chapter.

The rules for finding the first element are as follows:

(1) Boundary potential elements are given preference over potential elements which lie in the interior region.

(2) Lower boundary is given higher preference over all other boundaries. Preference then decreases going anti-clockwise along the boundary. Boundary corner points are lumped with the higher preference boundary.

(3) While searching for the potential elements lying on boundaries the following rule is observed: standing on the boundaries and facing the grid, the right-most potential element is given highest preference.

(4) While searching for the potential elements lying in the interior region of the grid the following rules are observed:

(a) A potential element lying on lower horizontal grid line is given preference over those lying on higher horizontal grid lines.

(b) Standing on the origin and looking along y-axis, right-most potential element is given highest preference.

Step 2: Once first potential element has been found, cells are formed about it. If two cells can be formed about a potential element then search is made first in the lower cell. If search is a failure there then search is made in the upper cell. If a point is found on a horizontal potential element (and on vertical potential element in case of left and right boundaries) then this element is deleted from the store.

After completing a contour it is seen if any potential element is remaining in the store. If it is so, it means another loop of the same

contour value is existing. This loop is searched in the same manner as stated above.

Step 3: For finding the exact location of contour on the potential element various interpolation methods can be used, the simplest being linear interpolation. In the present package linear and cubic interpolation facilities are available.

Cubic interpolation is done in the following manner : suppose ab (Fig. 2.1) is the element which is intersected by the contour. Using four more points, interpolation at n points (This number is to be chosen by the user of the package) between a and b are estimated. Now using these n+2 (a, b and n estimated points) points inverse interpolation is done to find the exact location of the contour value.

Inverse interpolation is possible only if function values at n+2 points are either increasing or decreasing monotonically. In most cases such a thing will happen except when situations of the type shown in Fig. 2.2 are faced: such outcomes of curve fitting are quite natural. In such cases only those estimated points are

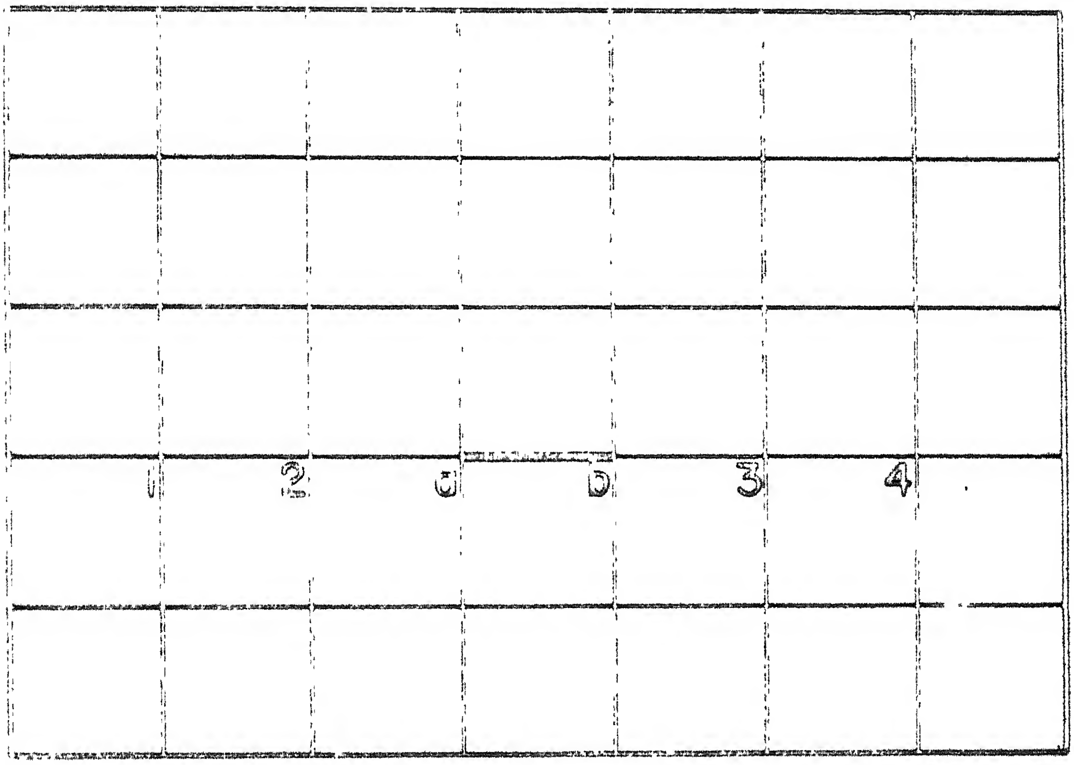

Fig.2.1 Example of a potential edge, ab

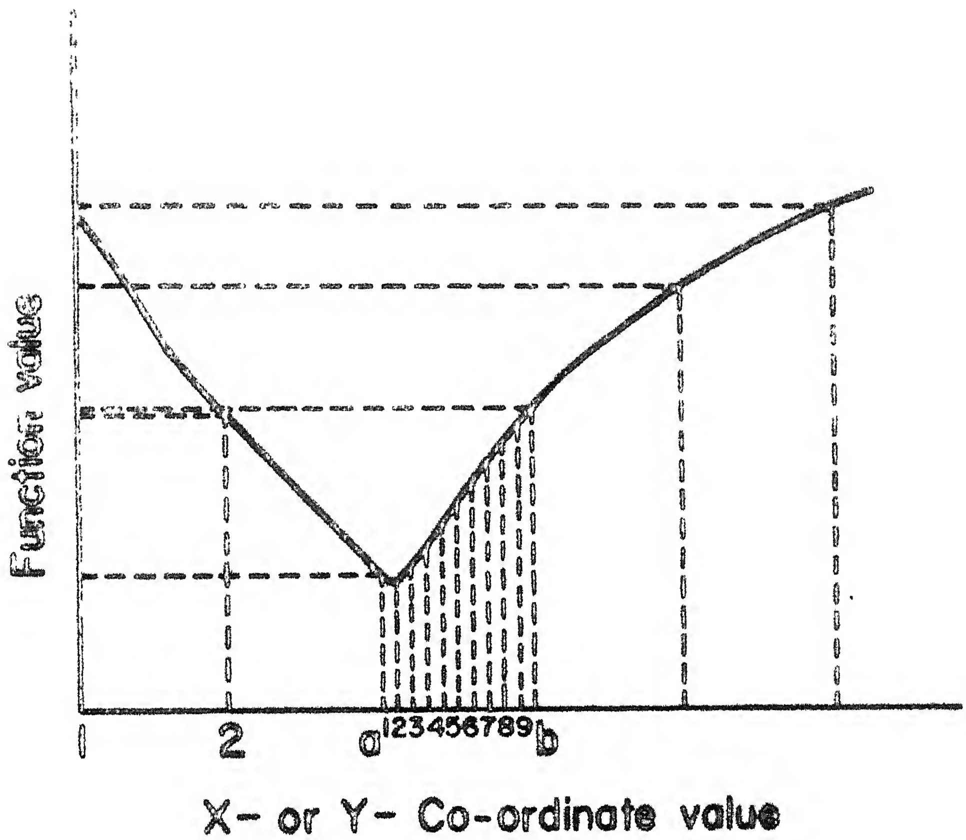

Fig.2.2 Cubic interpolation

considered which follow the expected trend and others are neglected. For example in Fig. 2.2 only 3,4,5,6,7,8,9,b are considered for inverse interpolation. There is no extra approximation involved in such an arrangement. This method of interpolation has its own drawbacks if curve fitting through the six points oscillates. In the present package, the subroutine given by Hiroshi Akima [5] has been used for cubic interpolation.

Step 4: Having found all the points at which contour intersects the grid a smooth curve is fitted through these points. To this end the subroutine given by Hiroshi Akima [5,6] has been used. This programme gives very good results.

Step 5: For displaying contours on graphics terminal LINE3 and MARKER subroutines of General purpose Graphic system (GPGS-001) have been used. Depending on accuracy needed one can choose the number of interpolation (n) points between two contour points lying on the grid. The interpolated points will be uniformly spaced. Irrespective of spacing of grid points

of the contour, the number of interpolated points between two grid points of contour will always be n all along the contour.

Chapter 3

DEGENERACIES

3.1 Different Types Degenerate Cases

It has been described by Cottafava and Moli [3] that a situation in which only three edges of a cell are potential elements is not possible. For nondegenerate cases two edges of the cell are potential elements. However all the four edges of the cell can be potential. In that case the cell is said to be degenerate. Through a degenerate cell the contour lines can have three equally possible arrangements. These have been shown in Fig. 3.1

It has been pointed out in [3,7,8] that there is not much to choose among these three choices. Unless there is some other imposing condition, one choice is as much valid as the other two. Most authors have dealt with such degeneracies by choosing the contour arrangement shown in Fig. 3.1c as the only possibility.

The possibility shown in Fig. 3.1c can occur only if the function has a maximum or a minimum or a saddle point in the cell. Then contour value should

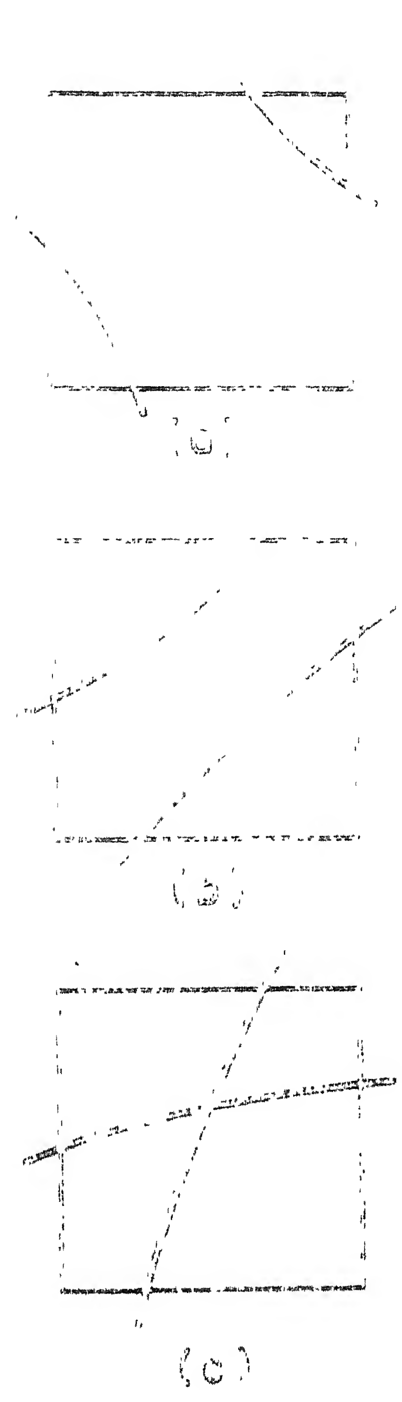

Fig.3.1 Alternative case of degeneraccies

be exactly equal to this maximum or minimum or saddle point value. Likelihood of such a coincidence is very low though not completely excluded. Moreover, in most situations if the choice shown in Fig. 3.1c is adopted, the shape of contours which come out are quite unrealistic. It is for this some authors have excluded choice in Fig. 3.1c with the justification stated above.

They have further tried to choose between Fig. 3.1a and Fig. 3.1b using some criteria. These criteria, again do not seem to be valid in general though they may be valid in particular applications [9].

Batcha and Reese [10] have solved the problem of degeneracy and thus its ambiguity by dividing the square cells into triangles by one or both diagonals. Degeneracy cannot occur in case of triangular cells.

Rothwell [10] after ruling out the possibility in Fig. 3.1c considers the possibilities in Fig. 3.1a and Fig. 3.1b and chooses the one in which the direction change is the least. Cottafava and Moli [3] on the other hand has suggested the strategy of shortest paths ; and if opposite points in the degenerate cell are equidistant then Fig. 3.1c is followed. It means degenerate cells in the grid have to be located and

stored before hand.

In order to obviate the problem of contour getting into an infinite loop (which owes to degeneracy) different authors have adopted different strategies. Almost all of these strategies are storage intensive and time consuming [3]. Some authors, on the other hand, consider such a possibility as very rare and thus have given them a secondary consideration [7].

In all the strategies proposed by various authors the degenerate cells have to be located, before the criterion suggested by these authors is applied to solve degeneracy. So while making the search in the cell (cell may be degenerate or nondegenerate) all the three edges of the cell have to be investigated for its being a potential element. This makes the methodology mentioned above rather slow even if there is no degeneracy.

In the present work, the possible case of Figure 3.1c has been excluded. Out of Figures 3.1a and 3.1b an attempt is made based on the rules given in Section 3.2 to select a particular configuration without attempting to resolve the degeneracy in a unambiguous manner.

3.2 Rules Governing Contour Search in a Cell

It is presumed in foregoing discussion that rules stated in Chapter 2 are still applicable. Following the rules set in Chapter 2 element ab (refer Fig. 3.2) will be the first element taken up from the store for a contour point. After this, edges of cell ac are searched for a contour point. To determine which of the remaining three edges of the cell is potential, a sequential search is made. Search is made according to the following set of rules. Once a potential element has been determined the contour point on it is located and the search proceeds to the next cell.

Rule 1: This rule holds if first potential element was lying on the boundary. Standing on the edge of the cell (on which a point has alrady been found) and looking into the cell, the three remaining edges of the cell are examined for their being potential elements going clockwise.

Rule 2: If the search for the contour was started with a potential element lying in the interior of the grid (that is not lying on the boundaries of the grid) then the three remaining edges of the cell are examined for their being potential elements going anti-clockwise.

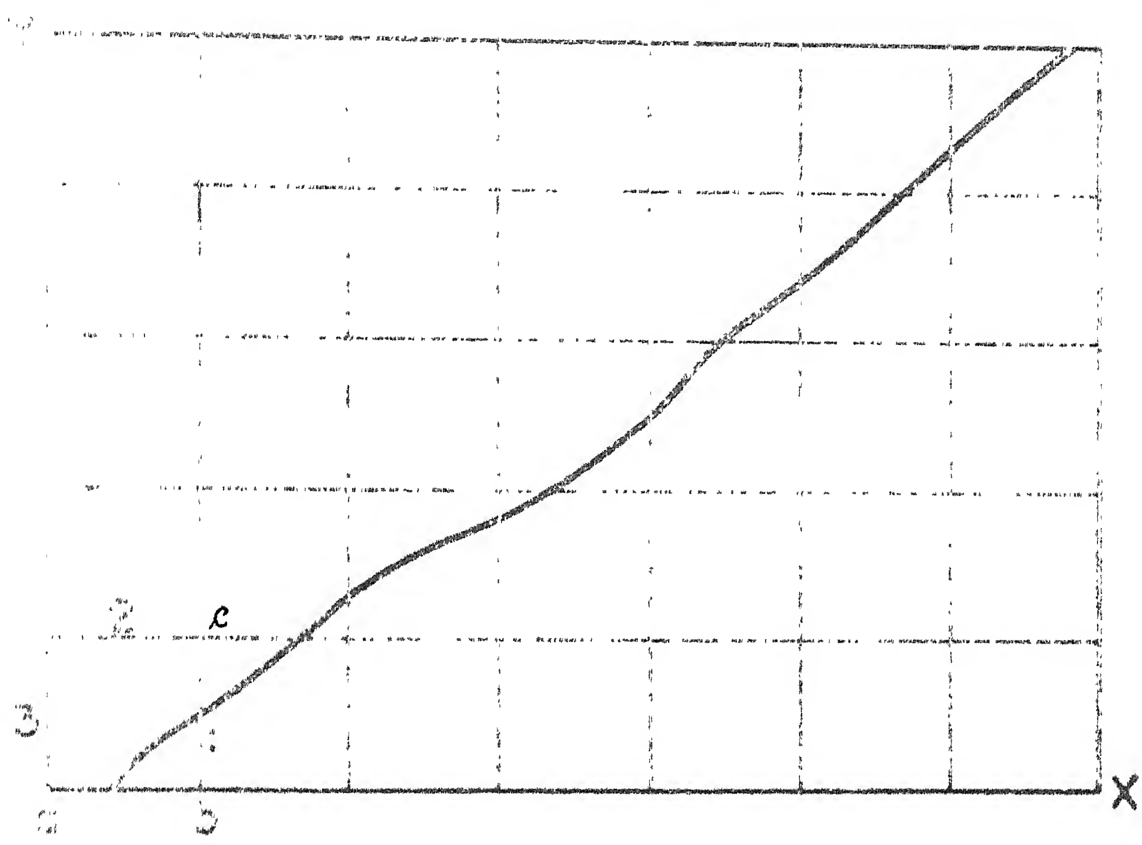

Fig. 3.2 Illustration of a transpassing contour

Rule 3: If there is a corner point, it is included in the edge of the cell encountered first in view of Rule 1 and Rule 2.

Rule 4: If a corner point is found and its horizontal potential element is not there in the store (it means that a point was found earlier also for the same contour value) then it is assumed that contour has ended. Further search for this branch is terminated.

This rule has relevance while investigating multi-branch contour starting from the same point.

Rule 5: If the search is a failure in the considered cell and the previously found point was a corner point then two remaining cells surrounding this corner point are searched in order for the possibility of contour point. The cell adjacent to the cell where the search is a failure is examined first. If search is a failure there also then contour is assumed to have ended.

Rule 6: If the search is a failure in the considered cell and the previous found point was not a corner point then the contour is assumed

to have ended. Such abrupt ending of contours can happen only in case of maximum and minimum contours as described by Morse [4].

3.3 Illustrative Example of Degeneracies

For the illustration of the stated rules the contour shown in Fig. 3.3 will be used as an example. Following the rules for choosing the first potential element one would choose ab. Then searching through cells one would reach point 7. According to rule 3, point 7 would belong to potential element cd. So the cell which is searched next is ci. Obviously here the search will be a failure. Hence according to rule 6 cell ck should be searched next where again the search fails. Therefore following the same rule search is made in the cell cf. Here the search will be successful. If the search fails here then it is assumed the contour has ended.

Cell cf is a degenerate cell. Now according to rule 1, 8 would be chosen as the next contour point (the third edge of the cell is not even examined). After completing the contour branch at 12 the store is looked into for any left out potential elements. In this the answer will be positive. Choice of the first potential element to start the contour is again made as per the

CENTRAL LIBRARY
I. I. T., Kanpur.
Acc. No. A 82649

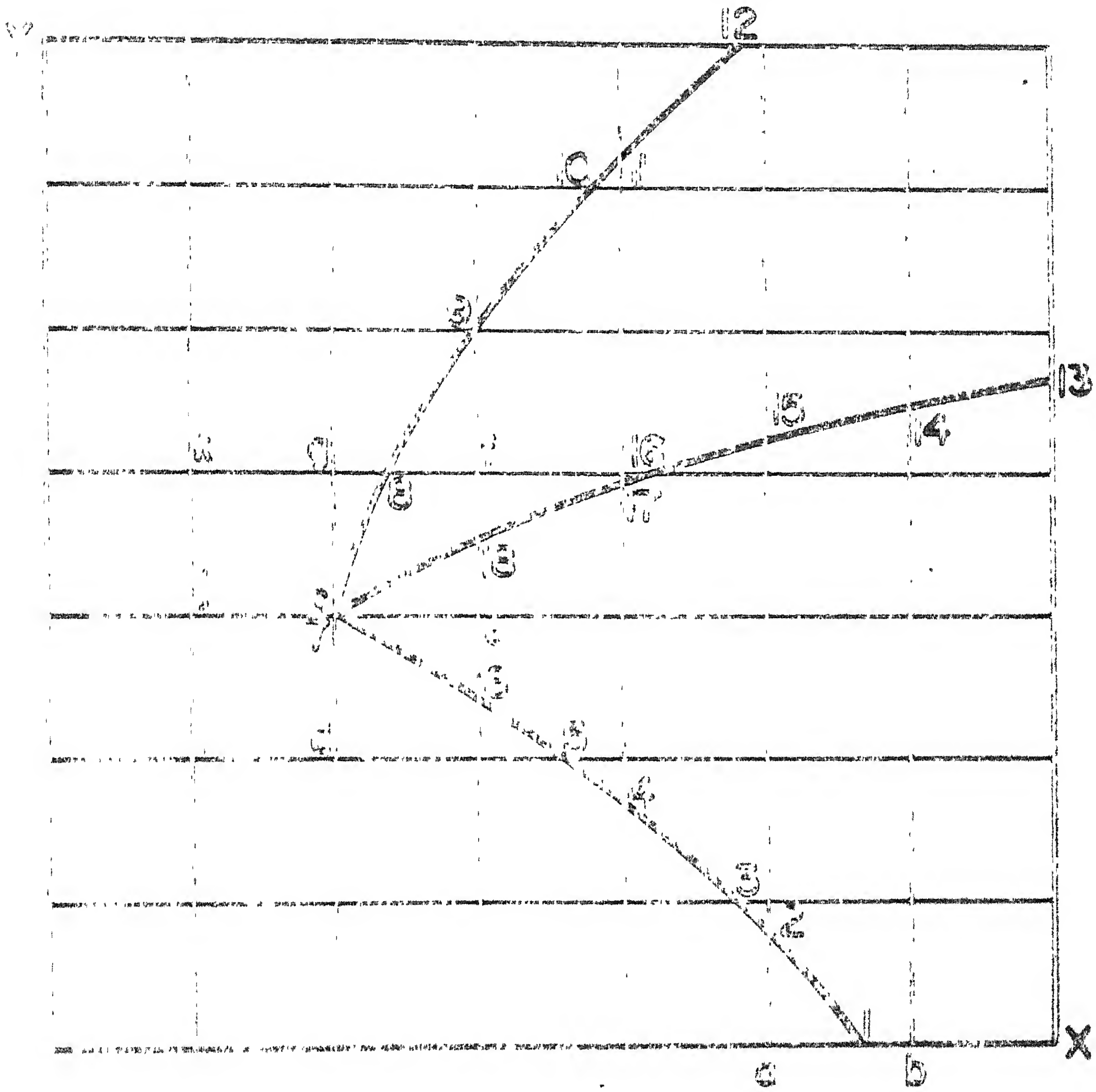

Fig.3.3 Illustration for contour search

rules stated in Chapter 2 regarding the choice of the first potential element.

So one starts from the point 13. After reaching the point 18 cell cf is to be searched which is degenerate. According to rule 1 point 7 is taken up as next contour point. But then horizontal potential element ce for this point is not there in the store. This is because it was deleted from the store when the same point was encountered while the first branch was being searched. Hence according to rule 4, contour is assumed to have ended.

3.4 Additional Illustrative Examples

Following the stated rules one can even draw contours of the type shown in Fig. 3.4. No branch of these contours is redrawn by the present package. These pathological examples of contours have been taken from [2].

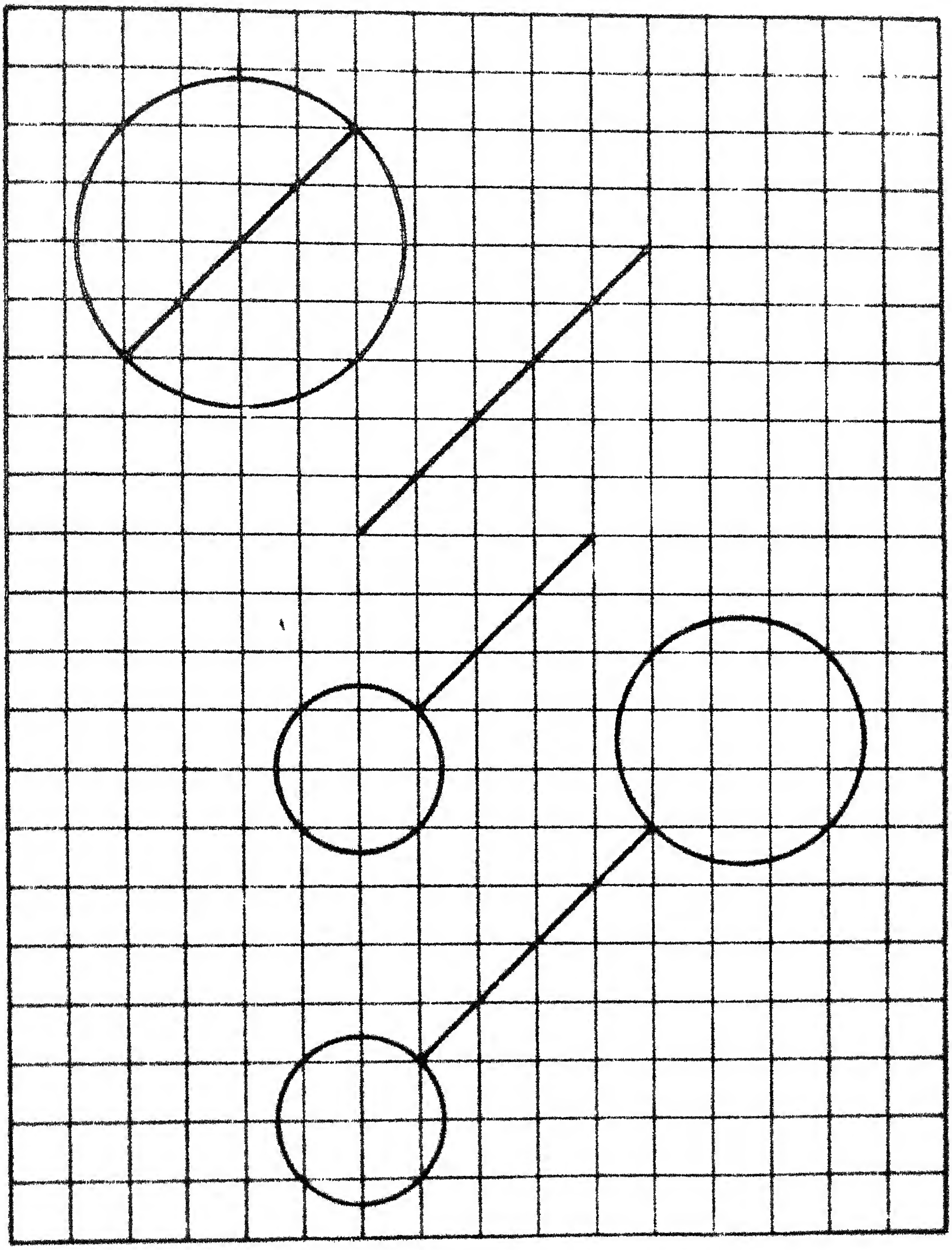

Fig. 3.4 Pathological examples of contours

Chapter 4

SALIENT FEATURES OF THE PACKAGE AND CASE STUDIES

4.1 Salient Features of the Package

Fig. 4.1 shows the flow chart for the contour search. In the flow chart only a broad outline of the method for contour search is available.

The programme can be accessed by the user by calling the subroutine CONPAC. The arguments of the subroutine and their description has been given in Appendix I. The values of the contours which have to be displayed can be supplied to the subroutine in two ways. One way is to supply them in an array as an argument of the subroutine CONPAC. The other way is to give the values of contours while interacting. In the latter case the contour values are evenly spaced starting from some supplied highest contour value (see Appendix I). The user can see exclusively a particular area of the grid by selecting an appropriate window. The number of grid lines to be displayed on each axis can again be chosen. The user can choose to avoid the grid altogether. The output of grid points and interpolated points of the contour can also be had if

FLOW CHART DESCRIBING GENERAL METHODOLOGY OF DRAWING CONTOURS

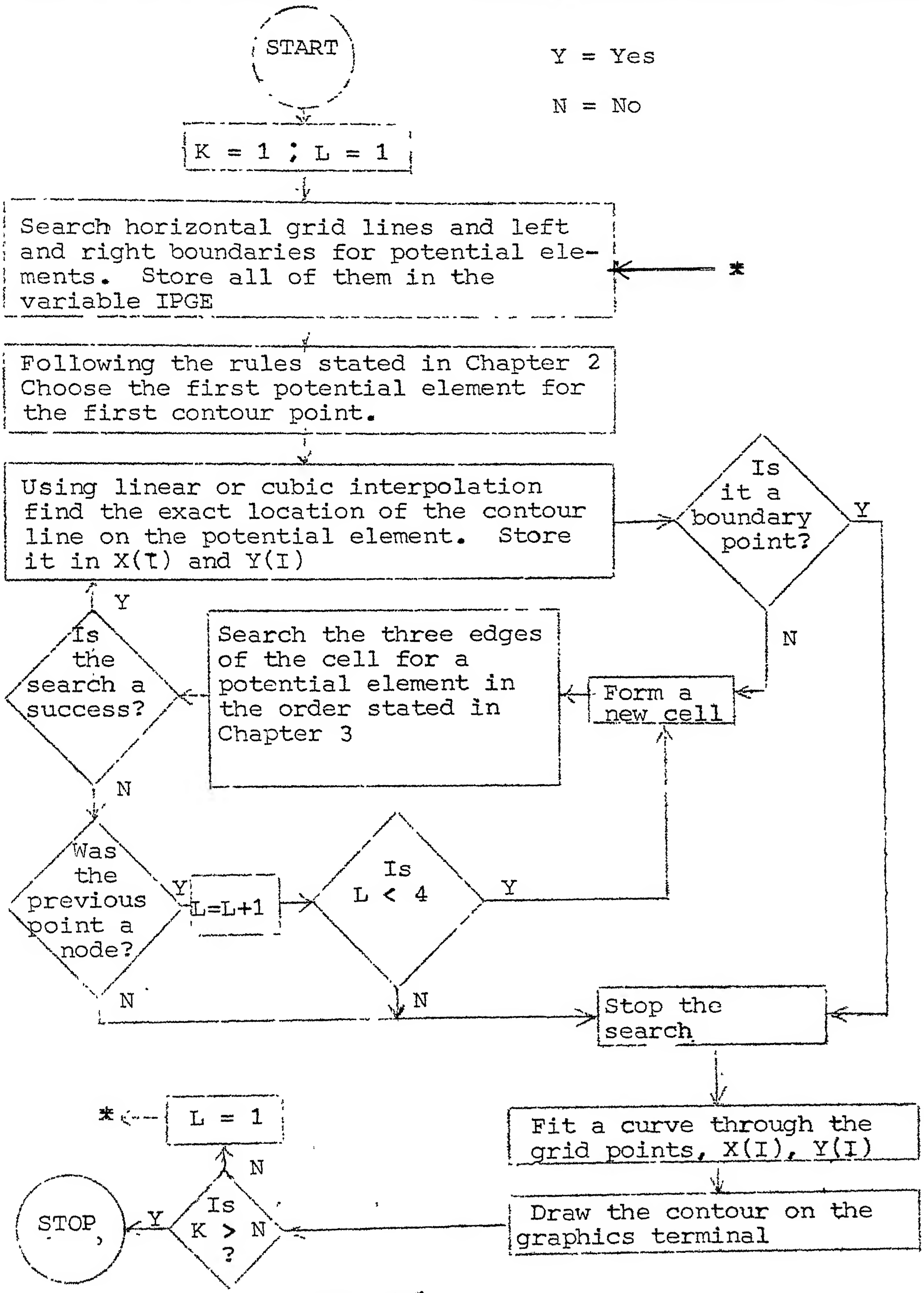

Fig. 4.1

the user wants.

4.2 Discussion on Case Studies

Two sets of data have been used to test the package. One set of data is related to Fermi surface and second set of data has to do with weather prediction. The Fermi surface data involves a mesh of size 40 x 40 and in case of weather prediction data mesh size is 49 x 32.

In both the cases contours have been drawn starting from some contour value and then decreasing this contour value by some constant amount. Otherwise package allows the user to give its own set of values of contours. The grid lines have been drawn for clarity.

There are degenerate cells in both sets of data. Some of these cells have been shaded in order to make them conspicuous. A contour which contains just one grid point has also been drawn in case of weather prediction data. The value of this contour is 5880. The weather prediction data plot also shows contours which pass through many consecutive nodes 5840.

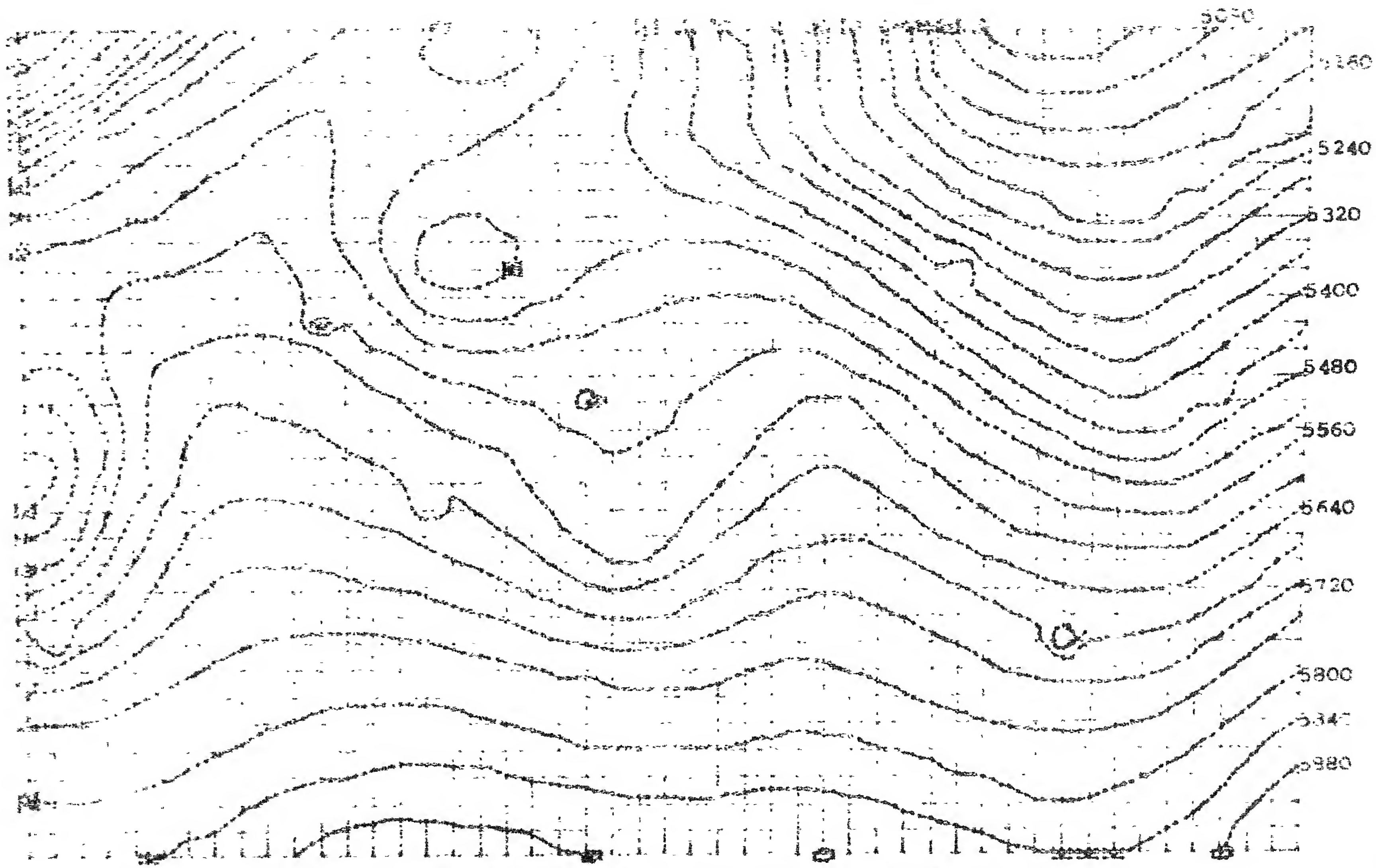

Fig. 4.2a Weather Prediction Data - First Set of Pressure Contours

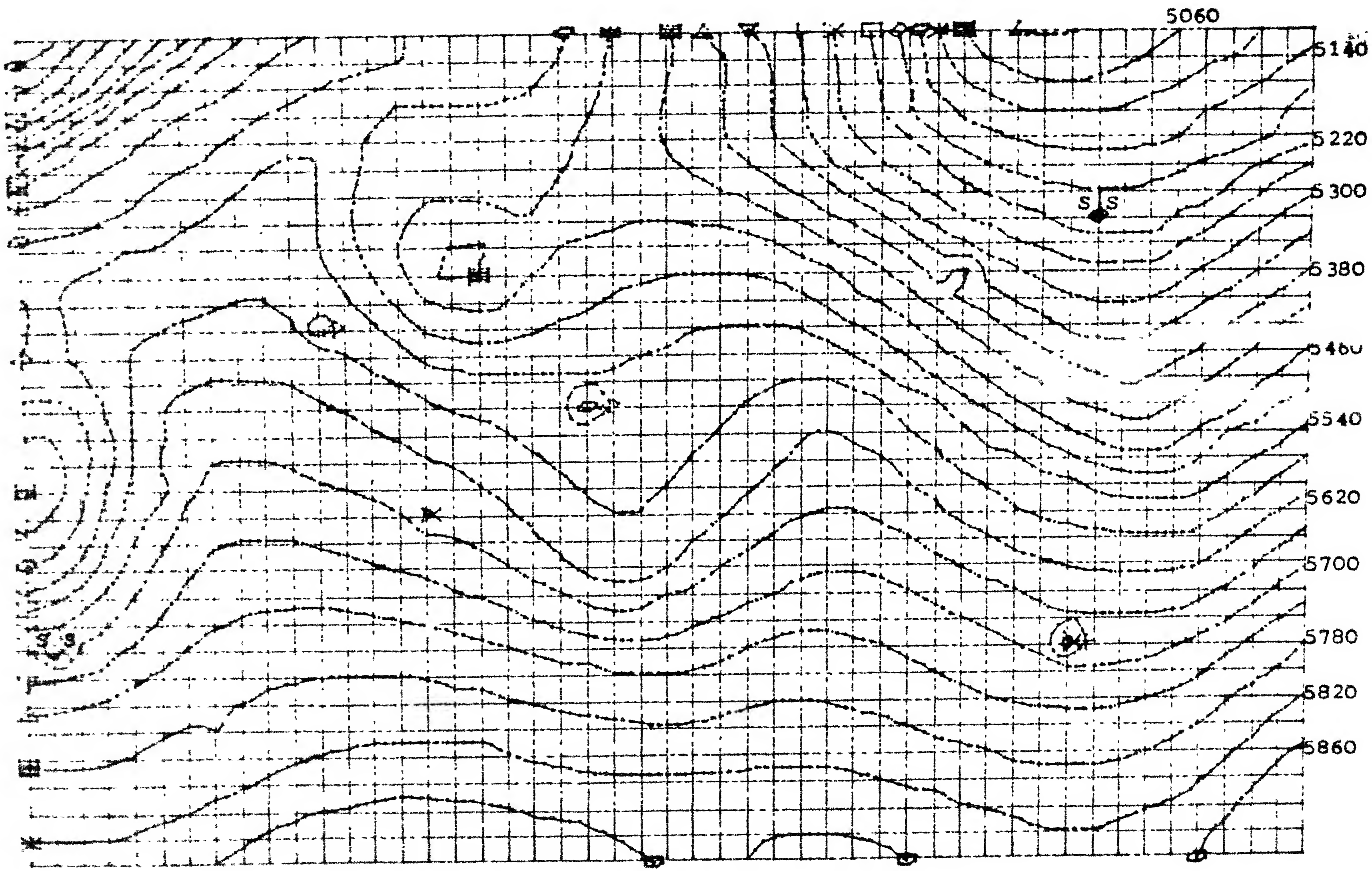

Fig. 4.2b Weather Prediction Data - Second Set of Pressure Contours

S = Shaded Cell

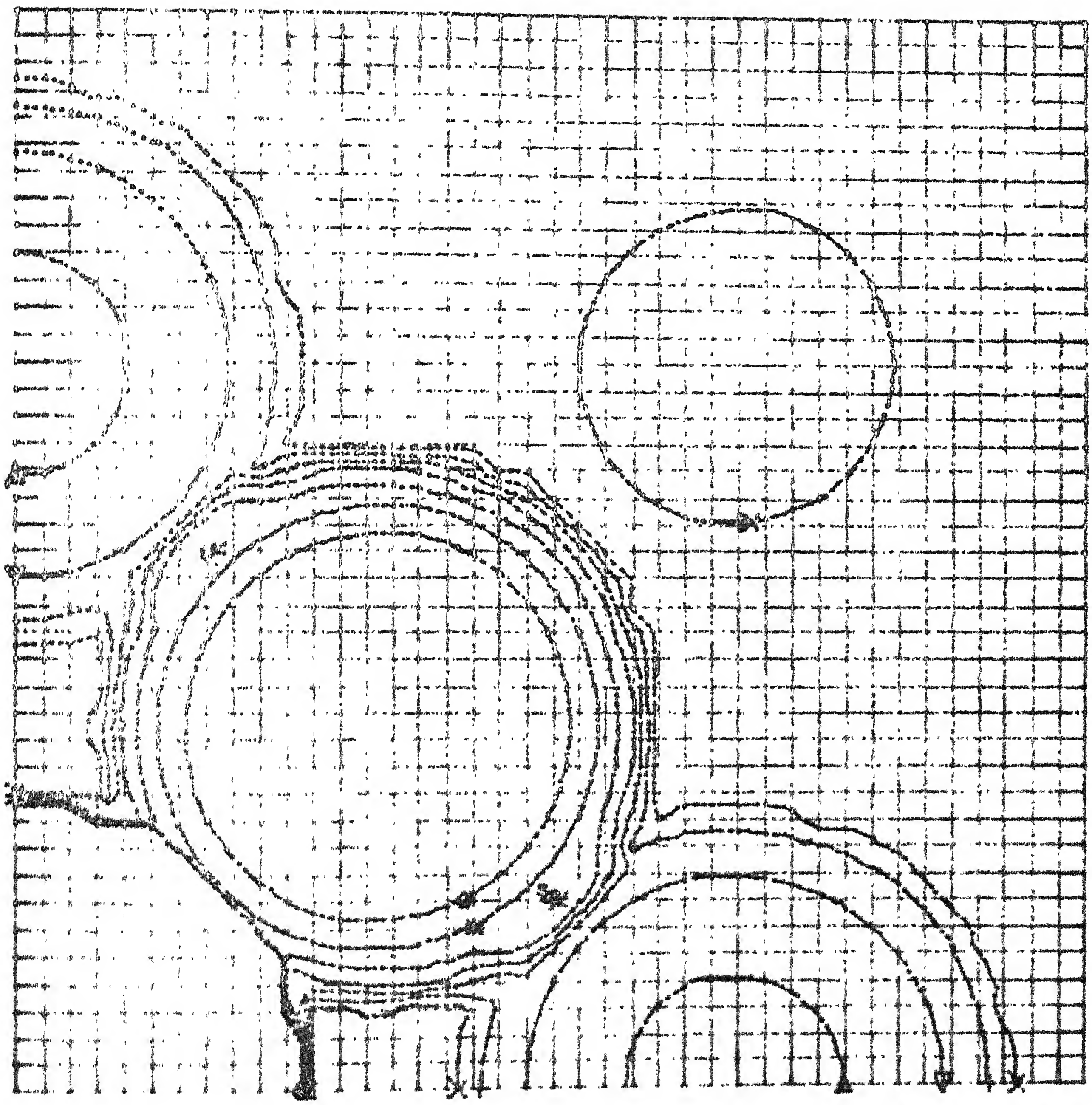

Fig. 4.3 Plot of Fermi Surface Data S = Shaded Cell

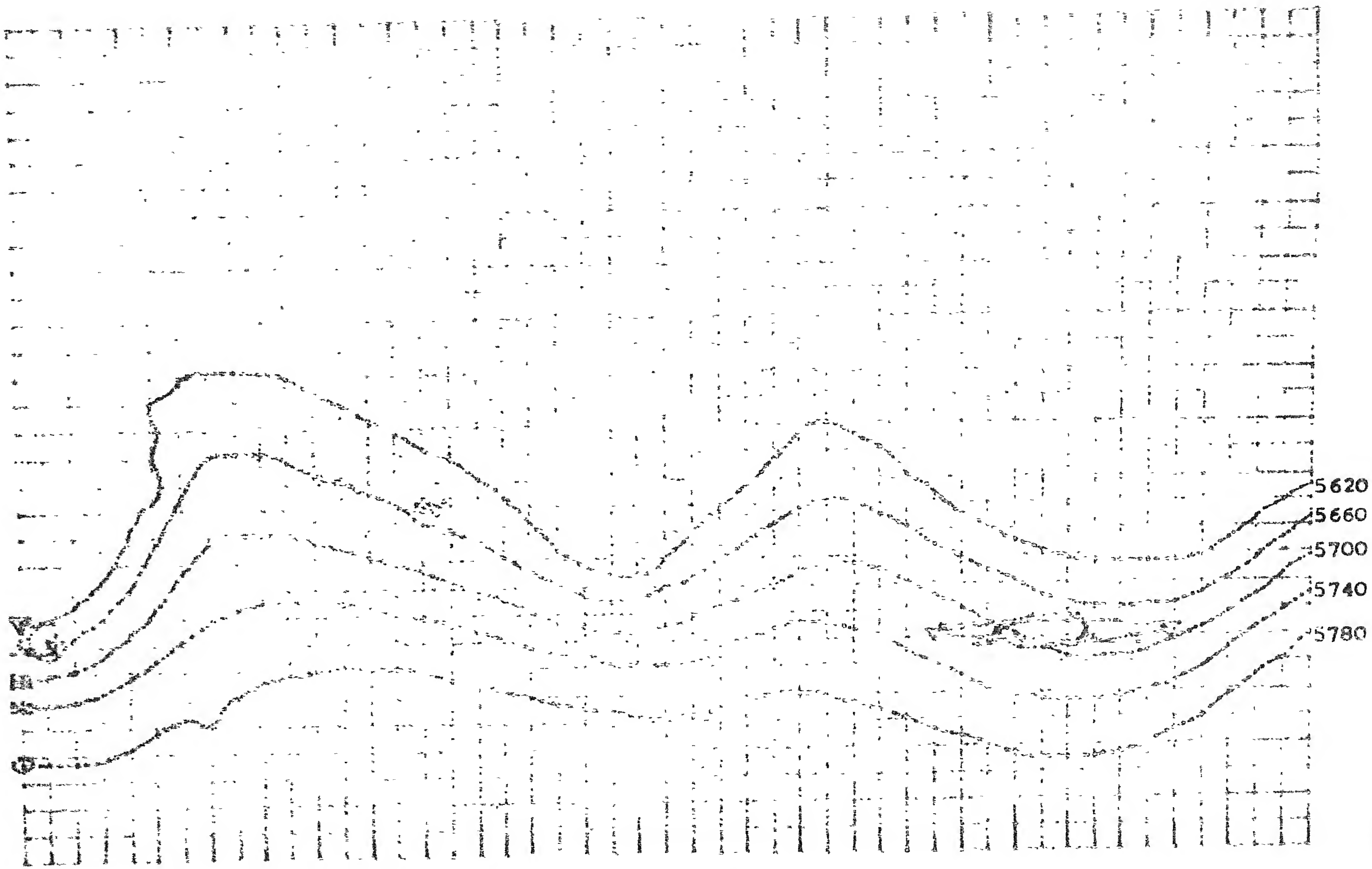

Fig. 4.6 Cubic Interpolation Failure

Chapter 5

RESULTS AND GUIDELINES FOR FURTHER WORK

5.1 Results

The present package can draw fairly good contours. The rules which have been developed in Chapters 2 and 3 have improved the efficiency of the search of contour grid points. For example, the situations where it can get into an infinite loop are automatically obviated without doing any extra calculations. Contours with bifurcating branches can also be drawn without reiterating any of its branch. In degenerate cases, by excluding the possibility in Fig. 3.1c, the intersection of contours is not allowed and this gives better shape of the contours. The pathological cases (Fig. 4.1) presented in his paper by Morse can be easily dealt with by the present package. This package can draw even those contours which consist of only one point. The subroutine used for curve fitting through contour grid points has also shown very good results.

The cubic interpolation on the potential element fails in those cases where contour loops are

very small. This has been shown in Fig. 4.4. When smaller window is given, even the portion of the contour which falls outside the window is computed. This makes the display of the contours rather slow. It would not be difficult to remove this drawback. Larger view of the contours can also be had by giving smaller values to maximum X- and Y-co-ordinates in the interaction. This will enable a closer view of the bottom-left corner of the grid only. In this case contours are not computed for the area which does not fall in the display region.

The package can also be extended to draw contours for the case when function values are available at irregularly distributed data points. Using the function values available at irregularly distributed points, function values at regular rectangular grid points can be interpolated. This can be done using the surface fit [12]. This interpolated data can be utilised to draw the contours using the present package.

5.2 Guidelines for Further Work

It is to be noted that the present package is not without shortcomings. There is still lot of scope for improvement on the present

package.

The reason of cubic interpolation failure while drawing very small loops of contours can be located and improved upon. Then redundant calculations done for contour parts falling outside the window can be easily eliminated.

The present work can be extended for the cases where grid is not rectangular; however there is some relation between the two position co-ordinates for example , a combination of radial and circular grid lines. This will require a small modification in the subroutine where interpolation is done on the grid element.

Then there is scope for accommodating the cases where grid boundaries are arbitrary, but the cells are rectangular. A suggestion which can be used to accommodate simple boundaries is shown in Fig. 5.1. The function values at grid nodes just next to the boundary are interpolated from function values available at grid nodes falling inside the boundary. The other points (indicated by crosses in Fig. 5.1) outside the boundaries are assigned much higher function values compared to function values available at grid points

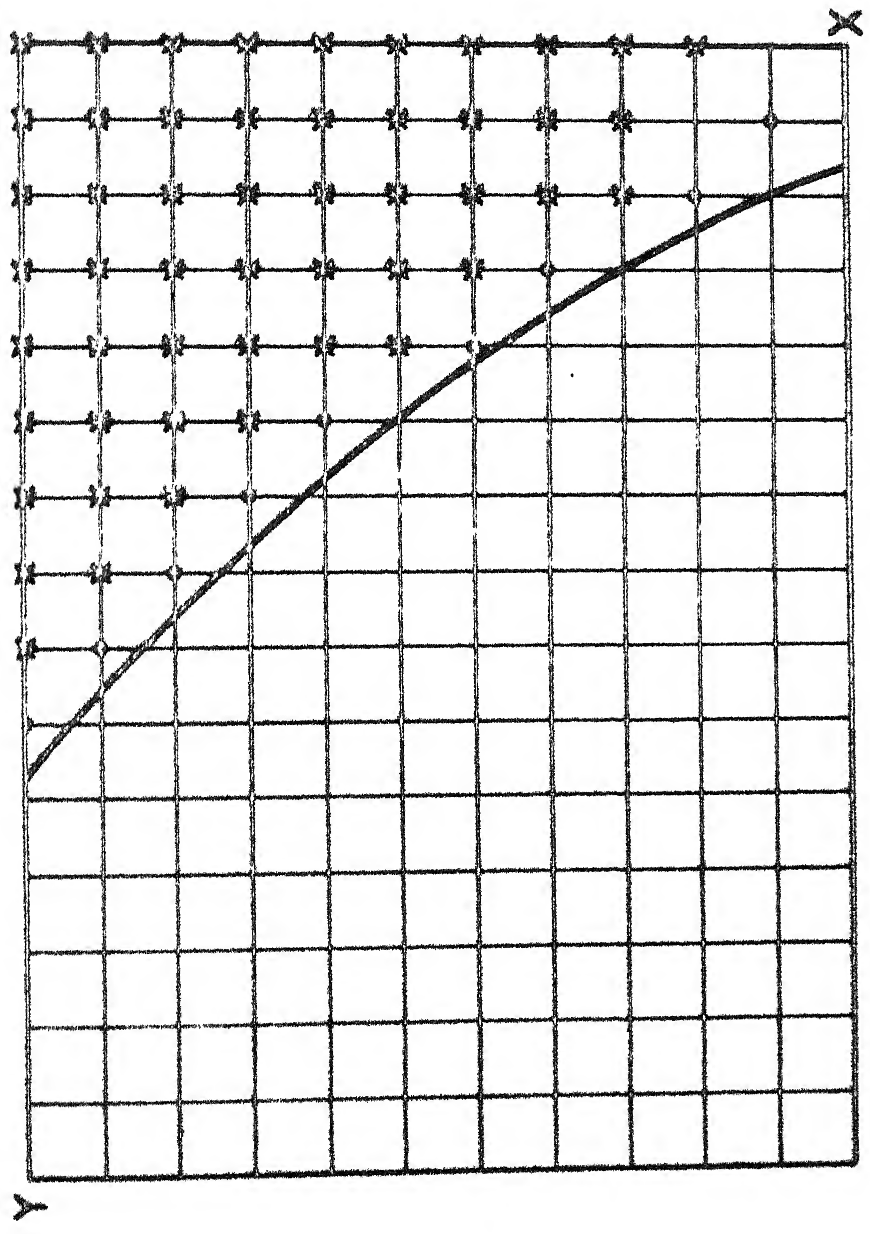

Fig. 5.1 Case of Arbitrary grid boundary

inside the boundaries. This arrangement definitely involves some distortion of contours near the boundaries. There is also a leakage of the contours outside the irregular boundary.

A major drawback of the proposed technique is its inability to resolve the case of a degenerate cell (Figure 3.1) unambiguously. The procedure adopted shows that it tends to generate multiply-connected contours whenever a degeneracy arises. This may not be the real situation in all the cases. To this extent, the package can be considered to have only limited applicability. It is necessary that in case of degeneracy, the data around a degenerate cell should be examined carefully and an appropriate configuration from Figs. 3.1a,3.1b and 3.1c should be selected unambiguously.

REFERENCES

1. Watkins, Steven L., "Masked Three Dimensional Plot Program with Rotations", Comm. of ACM, Vol. 17, No. 9, Sept 1974, pp 520-523.

2. Morse, Stephen P., " A Mathematical Model for the Analysis of Contour Line Data", J. of ACM, Vol. 15, No. 2, April 1968, pp 205-220.

3. Cottafava, G. and Moli, G.Le, " Automatic Contour Map", Comm. of the ACM, Vol. 12, No. 7, July 1969, pp 386-391.

4. Morse, Stephen P., " Concepts of use in Contour Map Processing", Comm. of the ACM, Vol. 12 , No.9, Mar 1969, pp 147-152.

5. Akima, Hiroshi., " Interpolation and Smooth Curve Fitting Based on Local Procedures", J. of ACM, Vol. 15, No. 10, October 1972, pp 914-918.

6. Akima, Hiroshi., " A New Method of Interpolation and Smooth Curve Fitting Based on Local Procedures" , J. of ACM, Vol. 17, No. 4, October 1970, pp 589-602.

7. Sutcliffe, D.C., " An Algorithm for Drawing the Curve $f(x,y) = 0$", Computer Journal, Vol. 19, No. 3, 1974, pp 246-249.

8. Crane, C.M., " Contour Plotting for Functions Specified at Nodal Points of an Irregular Mesh Based on an Arbitrary Two Parameter System" , Computer Journal, Vol. 15, No.15, pp 382-384.

9. Rae, A.I.M., "A Program to Contour Fourier Maps by use of an Incremental CRT Display", Acta Crystallography, Vol. 21, 1966, pp 618-619.

10. Rothwell, M.A., " A Computer Program for the Construction of Pole Figures" , J. of Appl. Cryst., Vol. 4, pp 494.

11. Akima, Hiroshi., " A Method of Bivariate Interpolation and Smooth Surface Fitting for Values Given at Irregularly Distributed Points", Report of U.S. Department of Commerce.

12. Mclain, D.H., " Drawing Contours from Arbitrary Data Points", The Computer Journal, Vol. 17, No. 4, Nov. 1974, pp 318-324.

13. Dayhoff, M.O., " A Contour Map Program for X-ray Crystallography", Comm. of ACM, Vol. 6, No. 10, Oct. 1963, pp 620-622.

14. Miller, C.L. and Laflame, R.A., " The Digital Terrain Model - Theory and Application", Photogrammetric Engg. Vol. 24, No. 3, June 1958, pp 433.

15. Young, W.H. and Isbell D.M., " Production Mapping with Orthophoto Digital Terrain Models", Photogrammetric Engg. and Remote Sensing, Vol. 44, No. 12, December 1978, pp 1521, 1536.

APPENDIX - I

DESCRIPTION OF PARAMETERS

THE INPUT PARAMETERS ARE :

IU = LOGICAL UNIT NUMBER OF STANDARD OUTPUT UNIT

NX = NO. OF GRID POINTS IN THE X-CORDINATE

NY = NO. OF GRID POINTS IN THE Y-CORDINATE

FVGP = DOUBLY DIMENSIONED ARRAY OF DIMENSION (NX,NY) STORING THE FUNCTION VALUES AT GRID NODES

N = NUMBER OF CONTOURS TO BE DISPLAYED

CV = ARRAY OF DIMENSION N STORING FUNCTIONAL VALUES OF CONTOURS TO BE DISPLAYED

THE OUTPUT PARAMETERS :

NCPL = MAX NO. OF INTERPOLATED POINTS THAT COULD BE ON ANY CONTOUR

X = ARRAY OF DIMENSION NCPL STORING X-COORDINATES OF THE CONTOUR POINTS

Y = ARRAY OF DIMENSION NCPL STORING Y-COORDINATES OF THE COUNTOUR POINTS

SOME VARIABLES USED INTERNALLY :

NCT = MAX NO. OF POTENTIAL ELEMENTS THAT COULD POSSIBLY LIE ON ANY HORIZONTAL GRID LINE

NPGE = ARRAY OF DIMENSION (NY+2) STORING NO. OF POTENTIAL ELEMENTS (< OR =NCT) ON DIFFERENT HORIZONTAL GRID LINES LINES

IPGE = DOUBLY DIMENSIONED ARRAY OF DIMENSION (NY+2,NCT) STORING POTENTIAL ELEMENTS (Dimension of IPGE can be reduced by avoiding drawing contours lying on fully horizontal grid lines)

INTERACTION PART

FOLLOWING INQUERIES ARE MADE WHILE IN INTERACTION MODE :

(1) TYPE MAXIMUM GRID NODES ON X & Y AXES
Accepted values are stored in IMAX & JMAX
NOTE : IMAX <= NX AND JMAX <= NY
If smaller values of IMAX & JMAX are given then lesser area of the grid is scanned for contours

(2) IF REQUIRED CONTOUR VALUES ARE BEING SUPPIED
THRU ARGUMENTS (N&CV) TYPE Y.
If anwser is not Y then following inqueries are made :
(a) TYPE NUMBER OF CONTOURS TO BE DISPLAYED
(b) TYPE THE LARGEST CONTOUR VALUE AND ALSO THE DIFFERNCE BETWEEN CONSEQUENT CONTOURS
Package in such cases will draw given number of contours starting with largest contour value and then with the given difference

(3) TYPE 3-D WINDOW SIZE
By varying the size of the window one can have closer view of some areas of the grid

(4) IF GRID LINES ARE NOT TO BE DISPLAYED TYPE N
If grid is to be displayed then follwing inquery is made:
TYPE SPACING BETWEEN TWO GRID LINES ON X & Y-AXES

(5) TYPE Y FOR PRINT OF CONTOUR POINTS

(6) TYPE 1 FOR LINEAR & 3 FOR CUBIC INTERPOLATION ON GRID ELEMENTS

(7) TYPE NO. OF INTERPOLATED POINTS BETWEEN TWO GRID POINTS OF THE CONTOUR
Grid points of the contour means points at which contour intersects the grid lines